PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum
Published by

Publisher:	Rector of the Ilmenau University of Technology Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor:	Marketing Department (Phone: +49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de) Faculty of Computer Science and Automation (Phone: +49 3677 69-2860) Univ.-Prof. Dr.-Ing. habil. Jens Haueisen
Editorial Deadline:	20. August 2010
Implementation:	Ilmenau University of Technology Felix Böckelmann Philipp Schmidt

USB-Flash-Version.

Publishing House:	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16 98693 Ilmenau
Production:	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts
Order trough:	Marketing Department (+49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher:	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

SECURE MULTICAST IN INTERNET-WIDE VPNS

Alexander Heinlein and Michael Rossberg and Guenter Schaefer

Ilmenau University of Technology

ABSTRACT

The shifting of sensitive communication into the potentially hostile Internet, led to the extensive deployment of virtual private networks (VPNs), allowing cheap, yet private information exchange over public infrastructures. Despite being standardized for more than a decade, there is no efficient solution for group communications in these scenarios, e.g., to distribute video messages, audio streams and software updates.

To overcome these shortcomings, we developed a system called **S**ecure **T**ransparent **O**verlay-**R**outed **M**ulticast (STORM) to realize a transparent, single-source multicast distribution tree for IP-multicast with the help of a number of distributed algorithms. It is able to rate nodes based on their age to efficiently integrate mobile nodes and make attacks on availability of the distribution service more difficult. Further metrics like delay, bandwidth, and hop count can be considered to construct and optimize efficient topologies. Moreover, tree balancing will occur to reduce latency and improve reliability as well as resilience against random failures or specific attacks. Simulative results demonstrated the adaptability of the topology towards various optimization goals.

Index Terms— Application Layer Multicast, Virtual Private Network, Peer-to-Peer, Security

1. INTRODUCTION

The last decade has led to a wide deployment of virtual private networks (VPNs) for the protection of cooperate and governmental networks. These VPNs form basically overlay topologies on top of the Internet, which may be either manually or automatically configured. As the manual setup of security associations between gateways turns out to be a complex and error-prone task that does not scale very well and is insufficient for mobile scenarios, there is a clear trend towards self-configuring networks that employ own functionality for routing, topology control, and node discovery.

However, the ability to influence topology and routing of VPNs automatically does not only simplify the deployment and administration. It also allows for the construction of distributed algorithms that adopt the overlay to environmental conditions such as denial-of-service (DoS) attacks. Within this context we

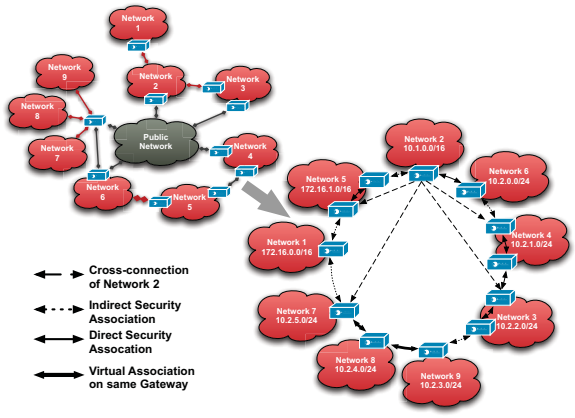

Fig. 1. Example of a dynamic VPN topology generated by SOLID

presented an approach called SOLID (Secure Overlay for IPsec Discovery) in prior work [1], which manages the task of automatic VPN configuration for complex IP security (IPsec) infrastructures by creating a ring structure where all gateways are arranged according to their internal IP address ranges. Gateway discovery and routing interact in SOLID to dynamically configure nested VPNs, like illustrated in Fig. 1.

Starting from this point, the novel flexibility does not only give the ability to simplify VPN configuration, but also to provide additional network services to the end-user. Thereby, the acceptance of additional costs and efforts to deploy VPNs will be further increased.

One of the perhaps best-suited services, to be deployed in VPNs, is IP layer multicast as it provides a mechanism for secure group communication, e.g. by distributing multimedia data or software updates. Furthermore, IP multicast is rather easy to adopt by end system programmers and may help to save scarce bandwidth, e.g., notebooks of traveling workers connected over a slow connection would no longer have to individually send a video stream to multiple receivers. Thus, within this article we present an approach **S**ecure **T**ransparent **O**verlay-**R**outed **M**ulticast (STORM) to distribute multicast over VPNs with a dynamic topology control.

STORM implements techniques and ideas from push-based application-level multicast (ALM) [2] to realize a fully transparent IP multicast by constructing

distribution trees with IPsec security associations. Already during the construction of the multicast trees different mechanisms ensure that resulting topologies are efficient as well as resistant to DoS attacks. Tree balancing is used to reduce latency and improve reliability in dynamic scenarios. Furthermore, all parts of the distributed algorithm are implemented within our configuration mechanism SOLID [1] in a real Linux environment as well as large scale OMNeT++ simulations.

The rest of the paper is structured as follows: the next section goes into a comprehensive set of goals for the distribution of multicast data in VPNs, followed by an analysis of the current state of the art. Section 4 covers the principles of STORM, and is accompanied by a discussion and evaluation in section 5. The article closes with a conclusion and brief outlook of planned future developments.

2. OBJECTIVES FOR VPN MULTICAST

The deployment of secure group communication is tied to the following functional goals:

1. **Generic Multicast Distribution:** The system should provide means to perform the delivery of packets to groups of VPN gateways and clients in inner networks respectively, allowing a multicast source to serve considerably more clients than possible by pure unicast.

2. **Transparency:** As an adaptation of existing IP multicast applications to a special VPN protocol, would imply considerable efforts, an IP multicast compatible interface must be provided, allowing clients to use common software to receive and send multicast packets.

3. **Self-Configuration:** The additional multicast service should keep the required amount of manual configuration at a minimum in order to be able to adapt to changes in the network environment and reduce administration costs.

Additionally, the following non–functional objectives should be fulfilled as good as possible:

4. **Scalability:** In order to distribute multicast streams even to large groups, the mechanism must be designed in distributed fashion, and every participating device must only be responsible for coping with an amount of control overhead that grows at most logarithmic with the total number of participants in the system.

5. **Dynamics & Robustness:** Multicast networks have to deal with regularly changing topologies as clients may subscribe to a group to peek at the content only, and leave shortly afterwards. Hence, care has to be taken to deal with fluctuating user behavior, and network properties such as delay and bandwidth.

6. **Efficiency:** To construct efficient distribution topologies, the following criteria have to be considered:

   (a) Latency: The delay between packet transmission and the delivery to the multicast members shall be as low as possible in order to allow a real-time service for multimedia data.

   (b) Network load: Not only for reasons of low latency, but also to reduce the network load, packets should be delivered using short paths and high bandwidth connections.

Due to the nature of the VPN environment, the fulfillment of the following security objectives against external as well as potential internal attackers is a primary concern:

7. **Confidentiality:** The unencrypted multicast traffic must be accessible only to legitimated VPN nodes.

8. **Data Integrity & Authentication:** Likewise, end-to-end data integrity and authentication must be guaranteed.

9. **Access Control:** This objective is twofold. First, only legitimate devices shall be able to access the multicast data, and after joining the multicast group regular reauthentications must ensure that the access capabilities are still valid. Second, the ability to send data to a multicast group shall be restricted to legitimate devices to prevent flooding attacks.

10. **Availability:** The primary focus of VPNs has been confidentiality, integrity, and authentication. However, with the increasing number of DoS attacks, the stable and resilient construction of overlay structures becomes a major concern. Thus, a multicast system must be realized by completely distributed algorithms in order to avoid single points of failure, and correct operation shall be guaranteed as long as a path to the multicast source is functional. In order to prevent DoS attacks to the multicast source, infrastructure hiding techniques need to be deployed.

3. RELATED WORK

The delivery of generic multicast to end-users over ALM has been widely discussed over the last decade

[3]. However, most approaches suffer from missing security mechanisms and are consequently not suited for secure group communications.

From a security perspective are specific VPN extensions better suited: The Internet VPN Group Management Protocol (IVGMP) [4] implements multicast routing on top of an IPsec-based VPN to provide security mechanisms. However, the VPN endpoints are controlled by a central device that is needed for configuration, key management, and group joins of VPN nodes, and thus questioning the scalability, robustness, and availability of the approach.

A ring-based approach is realized by VRing [5] with a self-organizing and distributed overlay. To compensate the long path lengths, an additional spare ring is introduced allowing a node to reach any other node with a maximum hop count of $2(\sqrt{N}-1)$ where N is the number of group members. However, for very large topologies this is still not feasible. Also, as packets arriving on the original ring need to be forwarded on both rings, additional link stress is introduced. Furthermore, security mechanisms have not been considered yet, but are planned for the future.

Dynamic Multipoint VPN (DMVPN) [6, 7] is a VPN auto-configuration system retailed by Cisco. It offers multicast distribution by deploying a rather static overlay network between hub gateways, which utilize Distance Vector Multicast Routing Protocol (DVMRP) to perform the actual multicast routing inside the VPN. Dynamic nodes must connect to one ore more hubs and simply retrieve multicast data from them. Drawbacks include complex configuration, low flexibility, and possible confidentiality problems as all hubs may eavesdrop all multicast data.

SOT (Secure Overlay Tree for Application Layer Multicast) [8] discusses the application of the Logical Key Hierarchy (LKH) [9] to ALM in order to create a scalable and yet secure multicast distribution scheme. However, the authors neither address availability issues nor problems caused by a real world implementation. Furthermore, the approach is not able to exploit a secure communication infrastructure like an approach specific for VPNs.

4. SECURE TRANSPARENT OVERLAY-ROUTED MULTICAST (STORM)

As the streaming of live multimedia data and quick delivery of status data, e.g., near real-time stock market information, are believed to be two major applications to multicast deployment in VPNs, source-to-client latency is a key issue. Thus, when considering the two fundamental implementation approaches, the creation of push- and pull-based topologies, the first ones offer a clear advantage as every packet is forwarded as quickly as possible, and no buffering has to take place.

The design space can be further narrowed by analyzing the communication behavior of the different multicast applications. Multimedia, status information, and software update data is usually provided by a single source, or at least a low number of sources. Other applications where every participant may transmit data to a group, e.g., DNS-SD [10] for local service announcement exist, but are considered a risk in VPN as multicast amplifies bandwidth attacks.

Hence, for VPNs it is the most adequate choice to construct distribution trees from each source node towards the subscribers and to secure the communication by individual security associations. This topology allows minimizing latency and is still efficient for a low number of sources. In the following subsections we will go into the assumptions, the optimization goals, and details of the construction process.

4.1. Assumptions

The setup of the distribution tree is based on the following two postulates:

The maximum bandwidth of a multicast flow is known: In order to optimally reserve resources for data forwarding in the peer-to-peer topology, it is required to know the flows maximum bandwidth. One the one hand a too low estimation will most likely lead to packet loss in forwarding VPN nodes, and on the other hand a too large estimation will make the distribution trees deeper, and thus increase latency. However, for high bandwidth multimedia streams it is easily possible to determine the exact bandwidth, and the other expected flows, e.g., for software updates, are usually of low bandwidth and can be estimated rather roughly.

Enough bandwidth in tree: To provide all clients with the multicast flows, the forwarding gateways need to have enough upload bandwidth. In contrast to open ALM services, this is not a large issue as participants have an interest to contribute to the closed VPN community.

4.2. Optimization Goals

Starting from the security and efficiency objectives, it is possible to derive the following goals:

Low tree height: In order to reduce latency and the chance of a failure of intermediate gateways, the length of the distribution paths and thus the height of the distribution tree shall be minimized. This is achieved by pulling nodes with high bandwidth into higher positions of the tree, i.e., to provide a higher fan-out, and by balancing the tree to better utilize bandwidth resources of the higher nodes.

Preference of elder nodes: Depending on the length of their sustained participation, elder and thus more stable nodes are to be placed in the higher regions of the distribution tree. This does not only prevent mobile or instable nodes to become important, but it may make it

also harder for a potential attacker to identify important other nodes.

Latency reduction: A further optimization goal is the exploitation of local distribution, i.e., neighbored nodes in the distribution tree should also be nearby in terms of latency.

With $\#succ$ denoting the successor count, $capacity$ the free bandwidth, and $c_{1...4}$ individual weighting factors, a set of successor nodes can subsequently be rated by the following function:

$$rating = c_1 \left(1 - \frac{\#succ}{\max \#succ}\right) + c_2 \frac{capacity}{\max capacity} + c_3 \frac{age}{\max age} + c_4 \left(1 - \frac{latency}{\max latency}\right)$$

This rating function is subsequently used to steer actual optimization procedures.

4.3. Topology Control Mechanisms

In order to create a distribution tree with the sketched properties, three mechanisms are used, utilizing the rating function. First, newly subscribed nodes will be placed in accordance to the previously defined optimizations goals. Second, periodic tree balancing assures an adaptation to changes in the transportation network and the VPN itself. Additionally, topology adaptations occur due to the failure of nodes.

4.3.1. Node Join Operations

After detecting multicast clients in their network, STORM gateways send a join message addressed to the multicast group along the structured overlay of the underlying VPN (see Fig. 2). Following the structure, the join message will either arrive at the multicast source after at most a logarithmic number of steps, or it passes another node that is already member of the searched group.

After reaching the source node or an intermediate node, this node decides whether it has enough free bandwidth for an additional child. If so, a security association is established to the requesting gateway and it becomes a child. Otherwise the discovered group member searches for another suitable position in the tree, according to the defined rating function. As children regularly report their free bandwidth and successor counts, it is immediately able to determine, if there is a suitable position in its own subtree. In this case the request message is iteratively forwarded to the best suited child. Alternatively, if there are not enough resources in the subtree, the request is passed up, until either a free spot is found or the source is reached. In the later case, there are no resources in the distribution tree at this time, thus violating the assumption of sufficient bandwidth and the request is rejected.

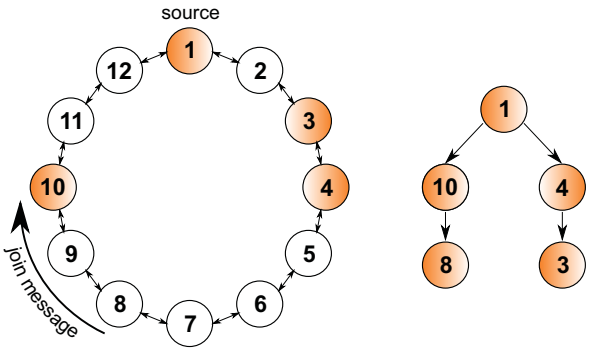

Fig. 2. Example of a node joining a tree

This join operation causes at most a logarithmic number of messages with regards to the number of VPN nodes as the first lookup step is bound to logarithmic steps and the second step of escalating or descending the distribution tree is bound to its height, which is also logarithmic due to the balancing operations, explained below. Furthermore, exploiting the topology aware structured overlay queries, chances are that found intermediate nodes are actually also close in terms of latency and topology of the transport network. And while it would be a conceivable alternative to register nodes with free bandwidth in the overlay structure, and not the source node itself, in order to release load from the higher regions of distribution tree, the chosen approach allows for better balancing properties.

4.3.2. Optimization for Latency & Robustness

Joining: As already sketched in the last section, when receiving a join message, nodes having no free bandwidth need to forward the request down one of their subtree if there is still free capacity left. They iteratively prefer the subtree with high capacity, low latency and high age, by weighting these factors using the rating function. However, to allow for a stable construction of distribution trees without frequent rebalancing operations, subtrees with considerable larger size must be preferred regardless of the rating.

Tree Balancing: The regularly exchanged successor count allows a parent to detect imbalances and perform explicit adjustments between subtrees by relocating children. As the other optimizations assure that only leaf nodes have spare capacity and to retain preceding optimizations, STORM attaches leaves of a subtree to leaf nodes of another subtree, allowing elder nodes to keep their higher positions. A configurable tree balancing rate allows to adapt to different optimization goals. Rate based balancing is achieved by calculating the optimal number of children per subtree, that is $perSubTree = \frac{totalSucc}{subTrees}$ where $totalSucc$ is the total successors count of all subtrees and $subTrees$ the number of subtrees. The tree balancing rate simply

specifies the maximum deviation from this number.

Load Optimization: If a gateway has enough free bandwidth for additional children, it may request more direct successors from its current children in order to relocate them to higher tree positions. Thus, latency is reduced and robustness increases. Requesting successors from a child with a high latency helps to reduce transmission delays and preferring high ages protects against attackers. The chosen child afterwards determines one of its successors to offer to the parent by utilizing the global optimization function

Position Switches: Parents regularly check if their children have a higher rating as themselves. If so, they change the position with the respective child to optimize the local tree. However, these parent initiated switches are very unlikely to occur except in highly diverse scenarios as the age of the parent and the number of total successors will always be higher compared to his children. Thus, only large differences in capacity or latency may lead to position switches, depending on the weighting of these factors.

4.3.3. Node Failure Detection & Repair

In order to perform node failure detection, associated VPN gateways already exchange periodic heartbeats. If gateways detect the absence of a heartbeat and its consecutive retransmissions, they will remove the corresponding security association. In case the failed node was a multicast child, the only further actions required, is the removal of the child from the successor list and forwarding tables.

In case the failed node is its multicast parent, searches for a new parents have to be initiated to reintegrate directly affected nodes and their children back into the tree. As all direct children of the failed node will search for a new group parent, measures have to be taken to ensure that they will not find nodes of affected subtrees; otherwise circular forwarding associations may emerge. To find a new parent, the same procedures as for the initial group join are used, that is, a join message is routed along the overlay structure. However, the join message is flagged to be a multicast recovery, ensuring that all gateways, including those that are already member of the searched group, will forward the message until it arrives at the multicast source. Next, the source sends the join message down the tree, until a free place is found at a gateway that is consequently none of the separated children. Other nodes are almost immediately informed of the new tree situation as nodes report important changes instantly to their parent without waiting for the next cycle. The time until reintegration may nevertheless be considerably long and depends on the interval at which heartbeat messages are exchanged, the diameter of the overlay structure, and on the height of the tree. But as only the parent of a separated subtree will have to be reintegrated, the number of exchanged messages is kept at a minimum and preceding optimizations of the subtree can be preserved. The costly reintegration supports the motivation to keep the tree height as low as possible and the most stable nodes close to the source.

4.4. Implementation

STORM is implemented in three components: the platform independent topology control, an OMNeT++/INET [11] library, and a Linux runtime environment. This approach allows for a comprising evaluation in simulations as well as real world scenarios. And while the simulation environment is crucial to discover large scale effects, the following section will focus on the discovered problems and solutions of the Linux prototype.

A rather often addressed problem is the estimation of the outgoing bandwidth of a node. Unfortunately, there are only two extreme solutions available: In a naïve solution a rather lengthy bulk transfer is performed towards a measuring host. A more elegant, but more unreliable solution is the transmission of short packet bursts and the consecutive measurement of packet interframe gaps [12]. Assuming no cross traffic, the measured dispersion will reflect bottleneck capacity. Thus, multiple measurements are required to limit these effects. Furthermore, STORM uses different corresponding nodes for its periodic measurements to eliminate effects, where the incoming bandwidth of the other side limits poses a bottleneck or a potential attacker artificially limits the measurements.

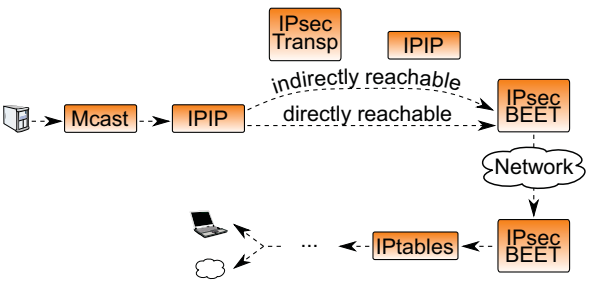

Fig. 3. Processing of multicast traffic

For supporting generic multicast applications, the Linux runtime environment manages the handling of Internet Group Management Protocol (IGMP) messages. To emulate the behavior of a conventional multicast router IGMP messages are intercepted by a raw socket, and the kernel's multicast API is utilized to encapsulate packets in IPIP-Tunnels and forward them to the successors. The traffic is individually encrypted at the gateways using Linux's native IPsec implementation in combination with strongSwan [13]. Fig. 3 illustrates the detailed packet processing.

To handle multicast traffic with IPsec, the packet originating from the multicast source and arriving at the gateway's virtual multicast interface is prepended by an IPIP header. Depending on the position of the child in the transport network, the packet may be additionally secured using IPsec's transport mode and an IPIP header for indirectly reachable successors. Consequently, by using a nested transport mode association, intermediate gateways cannot read or detect multicast traffic at all. Furthermore, STORM utilizes IPsec's Bound End-to-End Tunnel (BEET) mode to allow NAT traversal and finally transmits the packet on the network. The next gateway ensures with the help of iptables and dynamic firewall rules that the inner IP-addresses of the packet match the outer security association. Thus, STORM provide the same security level as tunnel-mode associations. Afterwards, the multicast traffic is forwarded to local receivers and remote gateways using the same procedure as before.

5. EVALUATION

Our evaluation is separated in two sections: starting with a qualitative discussion, quantitative, experimental results are given in the second part.

5.1. Qualitative Discussion

The following discussion is structured to reflect the arrangement of the objectives.

1. **Generic Multicast Distribution:** While currently still limited to IPv4 multicast, STORM allows for an application independent usage of multicast within the VPN.
2. **Transparency:** Towards client computers STORM gateways appear to be normal multicast routers, thus no changes to existing applications or operating systems are required.
3. **Self-Configuration:** All performed operations require no further configuration, other than the capabilities to provide access control. Additionally, the IP addresses of allowed multicast senders may be restricted in source gateways for security reasons.

From a non-functional view:

4. **Scalability:** Each gateway only stores information about its parent and direct successors. All operations are performed by using this information, and the introduced messages are only exchanged between neighboring tree levels with a maximum hop count that is limited by the tree height.
5. **Dynamics & Robustness:** When searching for a multicast group either a leaf is found or a node in the middle of the tree with an already exhausted bandwidth, as it would have otherwise requested children from one of the subtrees. Since exhausted nodes pass join requests down the tree, new nodes are always inserted as leaves. To keep more stable nodes in higher regions of the distribution tree, also the tree optimization algorithms take the node age into account, consequently newer or unstable nodes stay in the lower levels.
6. **Efficiency:** STORM uses regular topology checks that minimize the delay with the help of position switches. Additionally, other tree optimization algorithms also take the latency into account, providing different mechanisms to reduce it. The same mechanisms assure a low network load as a low latency also indicates nearby network devices. Furthermore, the locality aware search algorithm assures that local forwarders are more likely to be found.

From a security perspective STORM fulfills:

7. **Confidentiality:** The IPsec-based VPN provides encryption of all exchanged messages, ensuring no other than the two communication endpoints will be able to read the packet contents during transport through the overlay. In particular, no group keys are used in STORM.
8. **Data Integrity & Authentication:** IPsec also provides data integrity checks and authenticates each packet.
9. **Access Control:** Sending to a multicast group is restricted by the corresponding STORM gateway, which must in turn be registered in the VPN and authenticated by a certificate; otherwise multicast traffic will not be forwarded. Subscribing gateways also need to present a signed capability certificate to join a multicast group.
10. **Availability:** As VPN and multicast distribution are completely controlled by distributed mechanisms, node failures will not affect the remaining network as long as they do not interfere with the multicast source gateway. Selecting nodes, which have been contributing comparably lengthy and stable, for the upper levels of the distribution tree assures that successful DoS attacks either require a long setup time or have local effects only. Additionally, rate control mechanisms assure that the multicast system itself cannot be used to perform DoS attacks by potentially compromised participants.

Thus, all in all STORM accomplishes the set goals for VPN multicast from a qualitative point of view rather well. Further detailed studies are given in the following quantitative evaluation.

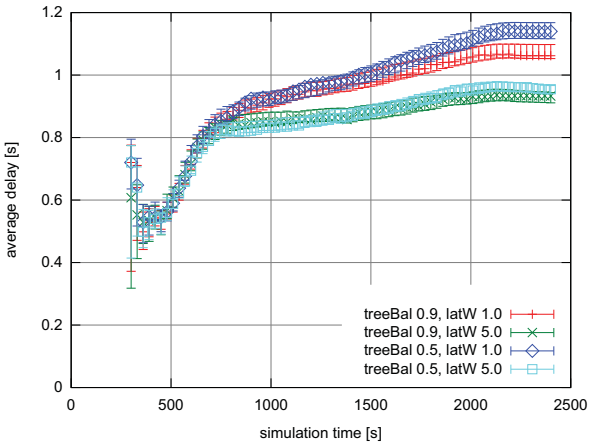

Fig. 4. Mean message delays at different configurations

5.2. Quantitative Results

Simulated networks are derived from data collected by CAIDA [14] to approximate actual Internet topologies and provide realistic test environments. In particular, OMNeT++ simulations were run in a network of 217 nodes with delays between neighboring gateways set to 100ms and between gateways and clients in their local subnetwork to 0.1ms. Each gateway has a bandwidth of 100 Mbit/s whereas the multicast source generates traffic at 25 Mbit/s representing an IPTV H.264 HiP Level 4 stream. To avoid measuring effects of the overlay creation, the multicast source gateway joined after 200 simulation seconds. Other Multicast gateways joined uniformly between 200 and 2000 simulation seconds. Unless otherwise stated, factor weightings for the rating function are 1.0.

5.2.1. Latency Optimization

For comparing influences of the rating function factors, different latency weightings were simulated to measure mean message delays between the multicast source and each client. Additionally, different tree balancing rates and their impact on message delays were evaluated. Fig. 4 shows four different configurations with latency weights 1.0 and 5.0 and tree balancing rates of 90% and 50%. As designed, higher latency weightings lead to decreased message delays for both tree balancing rates. By comparing different tree balancing rates, one can see that lower rates lead to higher message delays as individual tree branches will differ more in their height and hence forwarding path lengths will increase. In contrast, higher balancing rates will reduce the overall tree height as nodes are evenly distributed among the tree branches.

6. CONCLUSION & FUTURE WORK

Starting with the substantiation of the need for VPN multicast, this article motivated a detailed and comprehensive set of objectives for deploying this service by overlay networks. As current approaches were found unsuitable, our novel approach – called STORM – was presented. The qualitative and quantitative evaluation show that the entered direction is promising, and worth more detailed studies.

Hence, the focus of future research is planned to lay on the optional utilization of multicast support within the transport network, e.g., if some nodes are part of the same local area network. Furthermore, work needs to be done to study the interaction of multiple multicast groups in order to reuse already existing tree structures. The incorporation of a reputation system might allow to securely push badly performing participants into the lower regions of the distribution tree. In contrast to usual environment of reputation systems, it may be possible to take advantage of the tree structure and the fact that sybil attacks are not possible within VPNs.

7. REFERENCES

[1] Michael Rossberg, Guenter Schaefer, and Thorsten Strufe, "Distributed Automatic Configuration of Complex IPsec-Infrastructures," *Journal of Network and Systems Management*, vol. 18, pp. 300–326, 2010.

[2] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan, and Hui Zhang., "A Case for End System Multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456–1471, Oct 2002.

[3] Mojtaba Hosseini, Dewan Tanvir Ahmed, Shervin Shirmohammadi, and Nicolas D. Georganas, "A Survey of Application-Layer Multicast Protocols," *IEEE Communications Surveys and Tutorials*, 2007.

[4] Lina Alchaal, Vincent Roca, and Michel Habert, "Offering a Multicast Delivery Service in a Programmable Secure IP VPN Environment," in *In Fourth International Workshop on Networked Group Communication (NGC'02)*, 2002.

[5] Ahmed Sobeih, William Yurcik, and Jennifer C. Hou, "VRing: A Case for Building Application-Layer Multicast Rings (Rather Than Trees)," in *In Proceedings of the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)*, 2004, pp. 437–446.

[6] Scott Fluhrer, "SYSTEM AND METHOD FOR PROTECTED SPOKE TO SPOKE COMMUNICATION USING AN UNPROTECTED COMPUTER NETWORK," United States Patent US 2007/0271451 A1, 2007.

[7] Yusuf Bhaiji, *Network Security Technologies and Solutions*, chapter Part III: Data Privacy, Cisco Press, 1st edition, 2008.

[8] Wai-Pun Ken Yiu and Shueng-Han Gary Chan, "SOT: Secure Overlay Tree for Application Layer Multicast," in *IEEE International Conference on Communications*, 2004, pp. 1451–1455.

[9] Debby M. Wallner, Eric J. Harder, and Ryan C. Agee, "Key Management for Multicast: Issues and Architectures," IETF Request for Comments 2627 (Proposed standard), 1999.

[10] Stuart Cheshire and Marc Krochmal, "Multicast DNS," Internet Draft, IETF, 2010.

[11] András Varga and Rudolf Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008.

[12] Ravi Prasad, Margaret Murray, Constantinos Dovrolis, and kc claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE Network*, vol. 17, pp. 27–35, 2003.

[13] Andreas Steffen, "strongSwan - The new IKEv2 VPN Solution," Linuxtag 2007, Institute for Internet Technologies and Applications, Hochschule für Technik Rapperswil, Switzerland, 2007, `http://strongswan.org/docs/LinuxTag2007-strongSwan.pdf`.

[14] Young Hyun, Bradley Huffaker, Dan Andersen, Emile Aben, Colleen Shannon, Matthew Luckie, and kc claffy, "The CAIDA IPv4 Routed /24 Topology Dataset," 2010.